

# A Comparative Solution to ARGESIM Benchmark C4 'Dining Philosophers' with AnyLogic and MVSTUDIUM

Joshua Grove

Institute for Distributed Intelligent Systems, Saint-Petersburg State Polytechnical University,  
Saint-Petersburg, Russia; [jgrove\\_2000@yahoo.com](mailto:jgrove_2000@yahoo.com)

SNE 28(1), 2018, 31 - 33, DOI: 10.11128/sne.28.bn04.10406  
Received: February 20, 2012;  
Revised by EO August 20, 2017; Accepted: October 10, 2017  
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,  
ISSN Print 2305-9974, Online 2306-0271, [www.sne-journal.org](http://www.sne-journal.org)

**Abstract.** ARGESIM Benchmark C4 Dining Philosophers' is a very general one, and various modelling approaches are suitable. The presented solution compares modelling approaches with two similar simulators, but different in goals: AnyLogic and MVSTUDIUM. After a short introduction into the two object-oriented simulators, event-driven hybrid modelling approaches are presented – here the solution refers to the noticeable difference between the two modeling environments in messaging and signal processing. As option – and in order to prevent from deadlock – also a chopstick cleaning process is introduced, and the AnyLogic version provides a 2-D animation directly defined in the philosophers' objects. Finally some results for utilization and waiting times are presented – identically for both simulators, because of use of identical random streams.

## 1 Simulators

**AnyLogic** is an integrated graphical modeling environment from XJ Technologies. It is strongly Java based and allows for custom code, external libraries, and external data sources. Version 6.4.1 was used for this model. A 15-day demo trial with 40 different categories of examples can be downloaded from their website (<http://www.xjtek.com/>).

**MVSTUDIUM** is an integrated graphical modeling environment from the MVSTUDIUM Group. MVSTUDIUM is capable of modeling multi-component systems with continuous and hybrid processes. It is object oriented and based on UML (Unified Modeling Language) notation.

MVSTUDIUM does not require any traditional programming to build models. It allows for graphical programming and entering systems of second order differential, algebraic, and differential-algebraic equations. MVSTUDIUM includes wizards for easy 2D and 3D animation. Version 4.2 was used for this model. A 30-day demo trial version with ten working examples can be downloaded from the MVSTUDIUM website ([www.mvstudium.com](http://www.mvstudium.com)).

## 2 Modelling

In MVSTUDIUM, an object oriented event driven approach was taken for modeling the *Dinning Philosophers*. Two interacting hybrid classes were created. They were a table class and a philosopher class. In a similar way, two *Active Object Classes* were created in AnyLogic. The philosophers send requests to the table for sticks and the table responds by signaling if the request was successful or not.

A noticeable difference between the two modeling environments is the messaging and signal processing. AnyLogic allows for bi-directional connections between objects, where as a separate connection is required for MVSTUDIUM. In AnyLogic, sending messages with multiple parameters requires the programming of Java classes and adding code to connect the message to the state chart.

The *Table Class* created for the model contains a vector that indicates which of the five sticks are present or absent on the table at any time. Figure 1 and Figure 2 show the behavior chart, which is based on the UML state diagram, of a philosopher with the deadlock condition in AnyLogic and MVSTUDIUM, respectively.

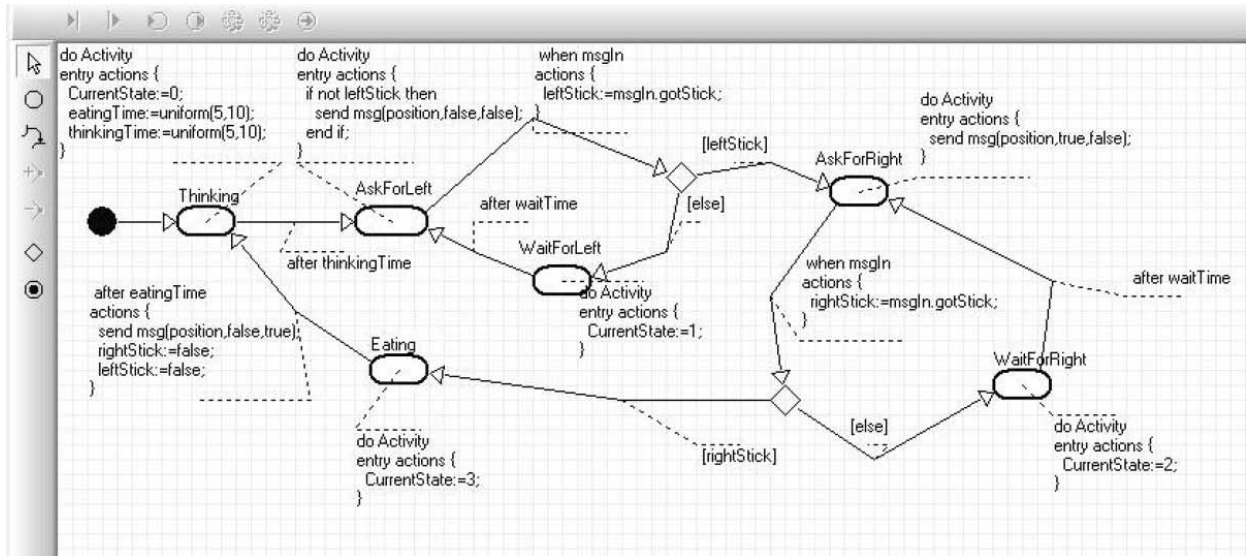


Figure 2. Behaviour Chart in MVSTUDIUM

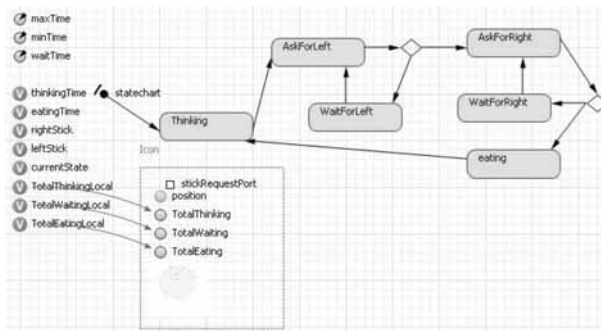


Figure 1. Behaviour Chart in AnyLogic.

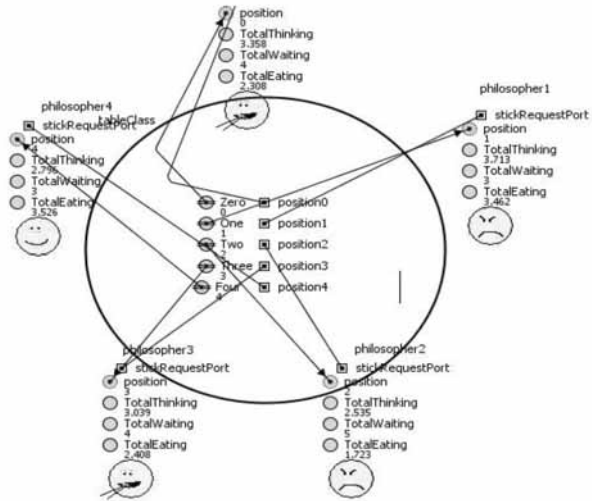


Figure 3. 2-D animation in AnyLogic.

The philosophers may starve or enter a deadlock condition, if they all grab a left stick before a philosopher can obtain a right stick. By adjusting the eating and thinking times a deadlock can become more probable. For instance a deadlock is more likely with a eating and thinking time at randomly selected in the interval of 0.5 to 1.5 than from an interval of 1 to 10 seconds.

For this comparison, three solution strategies were implemented in both modeling software solutions. They included

- a waiter that prevented utilization of all the chopsticks,
- a cleaning process that allowed communication between philosophers,
- and a put back strategy.

In AnyLogic, a 2-D animation was created directly in the philosopher class presentation to illustrate the philosopher's state (Figure 3).

In MVSTUDIUM, a similar animation was created in the testbench for each philosopher (Figure 5).

An integer value was used by both modeling software tools to indicate the state of the philosopher (thinking, waiting for sticks, waiting for one stick, and eating) and to switch the animation to the appropriate image (Figure 4).

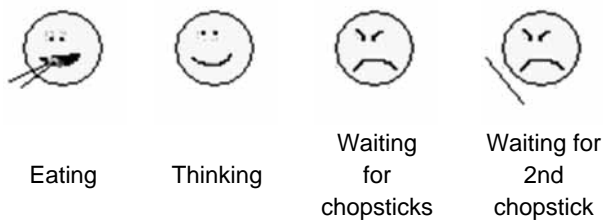


Figure 4. Status of philosophers in animations.

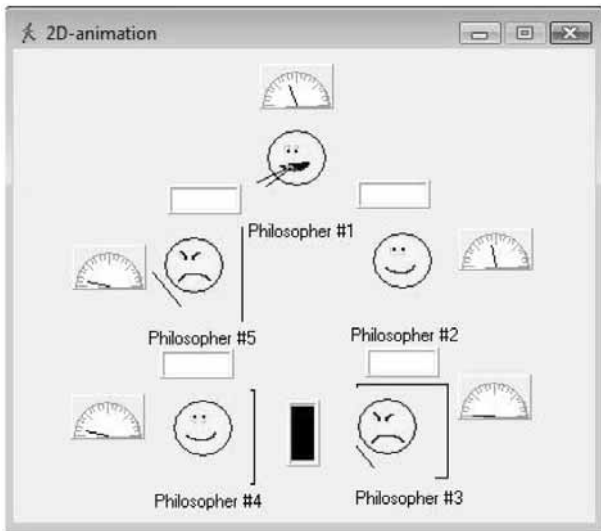


Figure 5. 2-D animation in MVSTUDIUM.

When creating the visual representation, the ability to create animations directly in the class in AnyLogic reduced the duplication of work by not having to create an animation for every instance of the philosopher class as required in MVSTUDIUM.

However, MVSTUDIUM 2D animation creator has a drag and drop style which is very easy to use.

### 3 Results

Table 1 shows the average times of the five philosophers in the waiting (hungry), thinking, and eating states for each of the different scenarios. The time for each simulation run was 100 time units.

Eating and thinking times were set randomly to be values between 0.5 and 1.5 time units. A waiting time of one time unit was used before the philosopher tried to obtain a stick after a failed attempt to obtain a stick from the table.

A deadlock was quickly achieved in the case with no strategy and the philosophers spent almost the whole time starving. The Put-back strategy achieved the longest time thinking and eating, followed by the waiter, and cleaning strategies.

The random eating and thinking time intervals were saved in a matrix and used for both AnyLogic and MVSTUDIUM. Subsequently, identical results were achieved from both.

	Deadlock	Waiter	Cleaning	Put back
Waiting	92.8	53.4	60.2	46.0
Thinking	3.7	23.1	19.8	27.0
Eating	2.9	22.8	19.7	26.7

Table 1. 2-Results AnyLogic and MVSTUDIUM.

